Congratulations! You passed!

Grade received 90% | Latest Submission Grade 90% | To pass 80% or higher

Go to next item

1. Suppose you learn a word embedding for a vocabulary of 10000 words. Then the embedding vectors could be 10000 dimensional, so as to capture the full range of variation and meaning in those words. 1 / 1 point

- ○ True
- ◉ False

Expand

Correct
The dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors range between 50 and 1000.

2. True/False: t-SNE is a linear transformation that allows us to solve analogies on word vectors. 1 / 1 point

- ○ True
- ◉ False

Expand

Correct
tr-SNE is a non-linear dimensionality reduction technique.

3. Suppose you download a pre-trained word embedding which has been trained on a huge corpus of text. You then use this word embedding to train an RNN for a language task of recognizing if someone is happy from a short snippet of text, using a small training set. 1 / 1 point

x (input text)	y (happy?)
I'm feeling wonderful today!	1
I'm bummed that my cat is ill.	0
Really enjoying this!	1

True/False: Then even if the word "upset" does not appear in your small training set, your RNN might reasonably be expected to recognize "I'm upset" as deserving a label y = 0.

- ◉ True
- ○ False

Expand

Correct
Yes, word vectors empower your model with an incredible ability to generalize. The vector for "upset" would contain a negative/unhappy connotation which will probably make your model classify the sentence as a "0".

4. Which of these equations do you think should hold for a good word embedding? (Check all that apply) 1 / 1 point

- ☑ $e_{boy} - e_{brother} \approx e_{girl} - e_{sister}$

Correct
Yes!

- ☐ $$e_{boy} - e_{brother} \approx e_{sister} - e_{girl}$$
- ☑ $$e_{boy} - e_{girl} \approx e_{brother} - e_{sister}$$

Correct
Yes!

Typesetting math: 100% $- e_{brother}$

Expand

Correct
Great, you got all the right answers.

5. Let A be an embedding matrix, and let o_{4567} be a one-hot vector corresponding to word 4567. Then to get the embedding of word 4567, why don't we call $A * o_{4567}$ in Python? 1 / 1 point

- None of the answers are correct: calling the Python snippet as described above is fine.
- This doesn't handle unknown words (<UNK>).
- The correct formula is $A^T * o_{4567}$
- It is computationally wasteful.

Typesetting math: 100%

Expand

Correct
Yes, the element-wise multiplication will be extremely inefficient.

6. When learning word embeddings, we create an artificial task of estimating $P(target \mid context)$. It is okay if we do poorly on this artificial prediction task; the more important by-product of this task is that we learn a useful set of word embeddings. 1 / 1 point

- True
- False

Expand

Correct

7. True/False: In the word2vec algorithm, you estimate $P(t \mid c)$, where t is the target word and c is a context word. t and c are chosen from the training set to be nearby words. 1 / 1 point

- True
- False

Expand

Correct
Yes, and are chosen from the training set to be nearby words.

8. Suppose you have a 10000 word vocabulary, and are learning 100-dimensional word embeddings. The word2vec model uses the following softmax function: 1 / 1 point

$$P(t \mid c) = \frac{e^{\theta_t^T e_c}}{\sum_{t'=1}^{10000} e^{\theta_{t'}^T e_c}}$$

True/False: After training, we should expect θ_t to be very close to e_c when t and c are the same word.

- True
- False

Expand

Correct

Correct
To review this concept watch the lecture.

9. Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings. The GloVe model minimizes this objective: **0 / 1 point**

$$\min \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij})(\theta_i^T e_j + b_i + b_j' - log X_{ij})^2$$

Which of these statements are correct? Check all that apply.

- [x] Theoretically, the weighting function $f(.)$ must satisfy $f(0) = 0$

  Correct

- [ ] $$\theta_i$$ and $$e_j$$ should be initialized to 0 at the beginning of training.
- [ ] $$X_{ij}$$ is the number of times word j appears in the context of word i.
- [x] $$\theta_i$$ and $$e_i$$ should be initialized randomly at the beginning of training.
- [x] $$\theta_i$$ and $$e_j$$ should be initialized randomly at the beginning of training.

Processing math: 100%

Expand

Incorrect
You didn't select all the correct answers

10. You have trained word embeddings using a text dataset of s_1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of s_2 words. Keeping in mind that using word embeddings is a form of transfer learning, under which of these circumstances would you expect the word embeddings to be helpful? **1 / 1 point**

- ( ) $s_1 << s_2$
- (•) $s_1 >> s_2$

Typesetting math: 100%

Expand

Correct
s_1 should transfer to s_2